# P3 Project

May 4, 2017

## 1 OpenStreetMap Project Data Wrangling with MongoDB

#### 1.0.1 *Dalal Alwedaah*

Map Area: Boston, MA, United States
https://mapzen.com/data/metro-extracts/your-extracts/3613179a3346

1. Problems Encountered in the Map
    - Over-abbreviated Place Names
    - multiple variables store in one name
    - Postal Codes
2. Data Overview
3. Additional Ideas
    - Contributor statistics and gamification suggestion
    - Additional data exploration using MongoDB
    - Conclusion

### 1.1 1. Problems Encountered in the Map

After initially downloading a small sample size of the Boston area and running it against a provisional python file(street_map_project_audit.py), I noticed three main problems with the data, which I will discuss in the following order:

- Incomplete street names ("Albany")
- Over-abbreviated place names ("Harrison Ave" and "88 E Newton St" )
- Inconsistent place names ("Starbucks","Starbucks Coffee" and "Starbuck's Coffee")
- Inconsistent bicycle rental names ("Hubway - Washington St. at Waltham St." and "Hubway - Tremont St / W Newton St" )
- Multiple variables store in one name("City Hair Salon & Body Waxing")
- Inconsistent postal codes ("MA 02118"and "02118")
- Inconsistent phone numbers ("(781) 749-3777", "+1 617 437 0300", "617-269-0110" and "6172478100")

### 1.1.1 Place Names

the data contain some of street names without street type. I suggest to update name with missing data such that “Albany” becomes “Albany Street”. I handle name abbreviations and inconsistencies by updating all substrings in problematic address strings, such that “88 E Newton St” becomes “88 East Newton Street”. for inconsistent place names like(“Starbucks”,“Starbucks Coffee” and “Starbuck’s Coffee”)they must replaced with standerd one “Starbucks”.

### 1.1.2 Multiple variables store in one name

Some of names have more than one variable in it. For example: “City Hair Salon & Body Waxing” contain tow variables: place name “City Hair Salon” and place amenity: “Body Waxing”. we remove amenity tag “& amp;” from the name then split it into tow variable name and amenity.

### 1.1.3 Postal Codes

Postal code strings posed a different sort of problem, forcing a decision to strip all leading and trailing characters before and after the main 5-digit zip code. This effectually dropped all leading state characters (as in “MA 02118”) and 4-digit zip code extensions following a hyphen (“02284-6028”).

### 1.1.4 Phone numbers

ther are many format for writing phone numbers which lead to Inconsistent phone numbers (“(781) 749-3777”, “+1 617 437 0300”, “617-269-0110” and “6172478100”). I suggest to provide standerd format for phone numbers and convert all of other format to it.

## 1.2 2. Data Overview

This section contains basic statistics about the dataset and the MongoDB queries used to gather them.

### 1.2.1 File sizes

boston_part.osm ......... 127 MB
boston_part.osm.json .... 145 MB

### 1.2.2

# Number of documents

```
> db.boston_part_data.find().count()
```

```
660656
```

# Number of nodes

```
> db.boston_part_data.find({"type":"node"}).count()
```

```
565712
```

# Number of ways

```
> db.boston_part_data.find({"type":"way"}).count()

94772
```

# Number of unique users

```
> len(db.boston_part_data.distinct("created.user"))

529
```

# Top 3 contributing user

```
> topcon = db.boston_part_data.aggregate([{"$group":
{"_id":"$created.user", "count":{"$sum":1}}},
{"$sort":{"count":-1}}, {"$limit":3}])

[{u'count': 359901, u'_id': u'crschmidt'},
{u'count': 138456, u'_id': u'jremillard-massgis'},
{u'count': 38187, u'_id': u'ryebread'}]
```

# Number of users appearing only once (having 1 post)

```
> db.boston_part_data.aggregate([{"$group":{"_id":"$created.user",
"count":{"$sum":1}}},{"$group":{"_id":"$count", "num_users":{"$sum":1}}},
{"$sort":{"_id":1}}, {"$limit":1}])

[{u'num_users': 142, u'_id': 1}]
```

# Most attribution

```
> db.boston_part_data.aggregate([{"$match":{"attribution":
{"$exists":1}}},{"$group": {"_id":"$attribution",
"count":{"$sum":1}}}, {"$sort":{"count":-1}}])


 [{u'_id': u'Office of Geographic and Environmental Information (MassGIS)',
  u'count': 44234},
 {u'_id': u'massDOT', u'count': 1740},
 {u'_id': u'Office of Geographic and Environmental Information (MassGIS),
 Massachusetts Emergency Management Agency',u'count': 40},
 {u'_id': u'USGS 2001 County Boundary', u'count': 33},
 {u'_id': u'Office of Geographic and Environmental Information (MassGIS),
 Commonwealth of Massachusetts Executive Office of Environmental Affairs',u'count':
 {u'_id': u'massGIS', u'count': 8},
 {u'_id': u'Office of Geographic and Environmental Information (MassGIS),
 Massachusetts Emergency Management Agency (MEMA)',u'count': 2}]
```

## 1.3 3. Additional Ideas

### 1.3.1 Contributor statistics and classification suggestion

The contributions of users seems incredibly skewed, possibly due to automated versus manual map editing. Here are some user percentage statistics:

- Top user contribution percentage (“crschmidt”) - 54.48%
- Combined top 2 users’ contribution (“crschmidt” and “jremillard-massgis”) - 75.43%
- Combined Top 3 users contribution - 82.72%

I think that “crschmidt” is the one who set the first map because its contributions accourres between 2007 and 2009. also you can notic that most of his node is basic node without any information about the names and address. the second one is “jremillard-massgis” at fist we an notic that he is related to “massgis” (the source of imported data) its contributions accourres in 2013. based on wiki page at Jan 2009, User crschmidt imported the 290,000 buildings around Boston. and at April 2013, the rest of the state MassGIS Buildings Import was completed.

when I explore the Most attribution (“crschmidt” and “jremillard-massgis” contributions are not included. just 46070 from 660656 - 7% of the whole dataset ) in this part from boston I found that ther are tow main attributions :

- Office of Geographic and Environmental Information (MassGIS) - 96.23% (from 7%)
- The Massachusetts Department of Transportation (massDOT) - 3.77% (from 7%)

MassDOT transform some of MassGIS data into streetmap site. I suggest to classify the creator into tow kinds: individual and organization. this will help people to be motivated to contreput and competing each other. puting tow sepreted leaderboard will reduce the gap between competitors.

### 1.3.2 Additional data exploration using MongoDB queries

**Top 10 appearing amenities**

```
> db.boston_part_data.aggregate([{"$match":
{"amenity":{"$exists":1}}},
{"$group":{"_id":"$amenity","count":{"$sum":1}}},
{"$sort":{"count":-1}}, {"$limit":10}])

[{u'_id': u'parking', u'count': 408},
 {u'_id': u'bench', u'count': 398},
 {u'_id': u'school', u'count': 246},
 {u'_id': u'restaurant', u'count': 149},
 {u'_id': u'library', u'count': 102},
 {u'_id': u'place_of_worship', u'count': 100},
 {u'_id': u'cafe', u'count': 65},
 {u'_id': u'fast_food', u'count': 54},
 {u'_id': u'fountain', u'count': 53},
 {u'_id': u'bicycle_rental', u'count': 47}]
```

### Biggest religion (no surprise here)

```
> db.boston_part_data.aggregate([{"$match":
{"amenity":{"$exists":1}, "amenity":"place_of_worship"}},
{"$group":{"_id":"$religion", "count":{"$sum":1}}},
{"$sort":{"count":-1}}, {"$limit":1}])
```

```
[{u'_id': u'christian', u'count': 92}]
```

### Most popular cuisines is italian and american

```
> db.boston_part_data.aggregate([{"$match":{"amenity":
{"$exists":1}, "amenity":"restaurant"}},
{"$group":{"_id":"$cuisine", "count":{"$sum":1}}},
{"$sort":{"count":-1}}, {"$limit":10}])
```

```
[{u'_id': None, u'count': 70},
 {u'_id': u'italian', u'count': 8},
 {u'_id': u'american', u'count': 8},
 {u'_id': u'pizza', u'count': 7},
 {u'_id': u'asian', u'count': 5},
 {u'_id': u'mexican', u'count': 5},
 {u'_id': u'japanese', u'count': 4},
 {u'_id': u'chinese', u'count': 4},
 {u'_id': u'thai', u'count': 3},
 {u'_id': u'international', u'count': 3}]
```

### Most popular fast food is Subway

```
> db.boston_part_data.aggregate([{"$match":
{"amenity":{"$exists":1}, "amenity":"fast_food"}},
{"$group":{"_id":"$name", "count":{"$sum":1}}},
{"$sort":{"count":-1}}, {"$limit":10}])
```

```
[{u'_id': u'Subway ', u'count': 5},
 {u'_id': u'Burger King ', u'count': 3},
 {u'_id': u"Jimmy John's ", u'count': 3},
 {u'_id': u"McDonald's ", u'count': 3},
 {u'_id': u'Boloco ', u'count': 2},
 {u'_id': u'Good Eats Pizza & Subs ', u'count': 1},
 {u'_id': u"Chuck and Ann's Submarines ", u'count': 1},
 {u'_id': u'Jamba Juice ', u'count': 1},
 {u'_id': u'Dunkin Donuts ', u'count': 1},
 {u'_id': u"Sullivan's ", u'count': 1}]
```

**Most popular cafe (starbucks is most popular if we treat inconsistent names)**

```
> db.boston_part_data.aggregate([{"$match":
{"amenity":{"$exists":1}, "amenity":"cafe"}},
{"$group":{"_id":"$name", "count":{"$sum":1}}},
{"$sort":{"count":-1}}, {"$limit":10}])
```

```
[{u'_id': u"Dunkin' Donuts ", u'count': 11},
 {u'_id': u'Starbucks ', u'count': 11},
 {u'_id': u'Starbucks Coffee ', u'count': 3},
 {u'_id': u'Temptations Cafe ', u'count': 2},
 {u'_id': u"Starbuck's Coffee ", u'count': 2},
 {u'_id': None, u'count': 2},
 {u'_id': u'Midway Cafe ', u'count': 1},
 {u'_id': u'Ula Cafe ', u'count': 1},
 {u'_id': u'Au Bon Pain ', u'count': 1},
 {u'_id': u'Dunkin Donuts ', u'count': 1}]
```

### 1.3.3 Conclusion

After this review of the data it's obvious that the boston area is incomplete and unclean. although the basis is good, marked places are limited (only 1.9% of the entier area). I perfer to apply updated system that encourage both indevigual persons and organization. I suggest to put standerd format and intrey modle that fource users to adding cleaned data. intery restriction may be a problem when using bots that must be updated to handle new standerde format. It also will be hard for indevsual to convert thir data to standerd format and time conseumaing. the system must be up to date for any change in the standerd format in each area. difficulties and restriction may causing pepole to migrate to other sites.